

Solving the Car Sequencing Problem with Constraint Logic Programming in the Automotive Industry

Thorsten Winterer
flexis AG, Stuttgart, Germany

November 17, 2011

flexis AG is a software company that specialises in planning and optimisation software for the automotive industry, with offices in Europe, North America, and Asia. One of our main products is a production sequencing solver that is used by several large truck and car manufacturers. The solver is based on Constraint Programming, and in its current incarnation it is implemented in ECLⁱPS^e.

The Car Sequencing Problem is the problem of finding a production sequence that fulfils a given set of hard constraints and is optimal with regard to some measure, usually a weighted number of soft constraint violations.

One of the earliest papers on the Car Sequencing Problem, by van Hentenryck, Simonis, and Dincbas, described a solver based on Constraint Logic Programming [3]. A general global sequencing constraint (among_seq in the Global Constraint Catalog) was introduced in [1]. Later papers considered improvements to the filtering algorithm for the sequencing constraint, e.g. [4]. In recent years the focus shifted to sequencing solvers based on Local Search. A review of the 2005 ROADEF challenge states that “*no constraint programming based method[s] were competing after the qualification phase*” [5].

However, there are several important differences between the Car Sequencing Problem as usually defined in the literature and what one will encounter in the automotive industry. The main difference is that in the industrial practice, the focus on the classic “car sequencing” constraint is too narrow. Customers may want to place orders with the same attribute, e.g., the same colour, into blocks of a given size. Some special configurations may only be built in certain shifts. A pilot order may only be built with a gap of specified length before the rest of the order. Some orders may have to be built in a given sequence, but not necessarily consecutively. A car sequencing solver needs to be able to model such constraints, and more. One paper that describes a variant of the Car Sequencing Problem with many real-world constraints is [2].

The second main difference is that the previous sequence must be taken into account, in order to achieve a feasible continuity. Continuity is essential, since car production often operates 24 hours per day, and even where it doesn’t, the production line is never emptied. The problem of continuity is usually not considered in the academic literature, though.

A third difference is the problem size. In the automotive industry sequences are often generated for a rolling horizon of between three days and one week.

For a truck manufacturer, this may mean up to 1,000 orders that need to be sequenced. For a car manufacturer, the problem may be an order of magnitude larger. The number of constraints varies, but can be up to one hundred.

Despite the large problem size, the solving process is done mostly interactively. The planner adapts the model to the current situation, possibly adding further constraints to test various scenarios, and then inspects the solution. The solution may be adapted manually, or some constraints are changed to test another scenario. Therefore, solutions must be computed quickly, within seconds for short or minutes for long sequences.

Due to the fact that the constraint models need to be adaptable, e.g., when new options are introduced, and that planners like to test different scenarios, must the modelling be flexible. We have implemented a standard set of high-level constraints that the planner can set and edit through a GUI, and which are then translated into low-level constraints that are posted in the sequencing solver. Enabling the planners to model the sequencing problem at their level of understanding is crucial for commercial success.

Furthermore, the sequencing solver core, including the generic model and the search strategy, must be general enough to cover any vehicle manufacturer's sequencing planning process. While a standard process may employ daily sequencing runs with a rolling time horizon, other processes use monthly planning runs with correspondingly larger problem sizes, while still other manufacturers sequence only the next 50-100 vehicles but need the sequencing solver to react to short-term disturbances in the supply chain. The model must also be easily extensible with custom constraints that are not part of the standard library but that are required by a customer. Since the models often differ even between production lines at the same factory, anything but a generic solver would create maintenance headaches.

One big advantage of CP over other solution methods is the stability of the search process: the solver will always produce the same solution for the same input data. Other methods that rely on randomisation, for instance Local Search methods, are at a disadvantage in this regard. We found that the sequence planners tend not to trust solvers that produce different results for the same input. Planners also often use a rather inefficient process with Local Search optimisers: let the solver run for some time, fix good stretches of the sequence, run the solver some more, unfix some stretches, etc. In effect, the planners restrict the solver to search around some local optimum, with the manual intervention taking up a long time.

These points (ease of modelling, ease of extensibility, and stability of search) led flexis to choose CP as methodology for its sequencing solver. The current solver is implemented in ECLⁱPS^e, which is embedded in a GUI that also connects the solver to the database. The solver does not use any standard constraints like the aforementioned sequencing constraint. Instead, constraints are implemented in a way that harmonises with the general search strategy that is used by the solver, in order to minimise unnecessary propagation.

The solver distinguishes between hard and soft constraints, and it can minimise soft constraint violations in a branch-and-bound search, but most customers prefer fast, good first solutions over waiting for an optimised solution that is often only marginally better. The solver therefore adapts the search strategy depending on the set of constraints in the model in order to guide the search towards a solution where as few soft constraints as possible are violated.

Our experience shows that a CLP-based approach is competitive for solving real-world sequencing problems. An earlier version of the flexis solver was tested on four of the problem instances published for the ROADEF'2005 challenge and matched the best solution in three cases, while coming close for the fourth. In a recent industrial project, the current version of the flexis solver used less than one minute to sequence 4,000 orders while considering 20 constraints.

We are currently working on integrating the sequencing solver with workload planning for the production lines.

References

- [1] N. Beldiceanu, E. Contejean: Introducing Global Constraints in CHIP. *Journal of Mathematical and Computer Modelling* 20(12), pp. 97–123, 1994.
- [2] M. Bergen, P. van Beek, T. Carchrae: Constraint-based vehicle assembly line sequencing. In *Proc. 14th Canadian Conf. on Artificial Intelligence*, LNCS 2056, pp. 88-99, 2001.
- [3] M. Dincbas, H. Simonis, P. van Hentenryck: Solving the car-sequencing problem in constraint logic programming. In *Proc. ECAI-88*, pp. 290-295, 1988.
- [4] W.-J. van Hoesve, G. Pesant, L.-M. Rousseau, A. Sabharwal: Revisiting the sequence constraint. In *12th Int. Conf. on Principles and Practice of Constraint Programming*, LNCS 4204, pp. 620-634, 2006.
- [5] C. Solnon, V.-D. Cung, A. Nguyen, C. Artigues: The car sequencing problem: overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem. In *EJOR* 191(3), pp. 912-927, 2008.